Amendments to the Specification:

Please see the attached amendments as noted on the specific pages and paragraphs of each addition.

Please replace the title shown on page 1 with the following title:

-- Health Trend Analysis Method on Utilization of Network Resources –

At page 13, line 10, change the paragraph to read as follows:

9. Unisys HealthEvents COM Object: A COM object that supports methods and properties describing a single health violation. A COM object is defined as a component written to the COM standard that provides interfaces which can be invoked by applications; see "COM Object". Glossary Item #40.

At page 13, line 18, change the paragraph to read as follows:

11. Health Events: Health issues detected by the Unisys HealthMonitor service. See "Health Issues". Glossary Item #10.

At page 13, line 20, change the paragraph to read as follows:

12. Health Violations: see "Health Events". Glossary Item #11.

At page 14, line 15, change the paragraph to read as follows:

19. Knowledge Scripts: The NetIQ name for scripts executed by their AppManager product. See "Server Director Scripts". Glossary Item #2.

At page 14, line 24, change the paragraph to read as follows:

22. HealthEvents COM Object: see "Unisys HealthEvents COM Object". Glossary Item #9.

At page 14, line 26, change the paragraph to read as follows:

23. Health Events Object: see "Unisys HealthEvents COM Object". Glossary Item #9.

At page 16, line 4, change the paragraph to read as follows:

34. HealthEvents dll file: see "HealthEvents dll". Glossary Item #332.

At page 18, line 1, change the paragraph to read as follows:

48.. HealthEvents Object: See "Unisys HealthEvents COM Object". Glossary Item #9.

At page 18, line 3, change the paragraph to read as follows:

49. Unisys-Supplied Policy File: A file of predefined XML statements describing health monitoring policies that is included with the HealthMonitor service. See "Canned XML". Glossary Item #21.

At page 18, line 17, change the paragraph to read as follows:

52. "Common Sense" Check: validation of a statistical trend against the relationship between the first and last data points used by the trend algorithm. For example, to validate a positive or upward trend, the value of the first data point should be below the value of the last data point. The "common sense" value is designated as "C".

At page 20, line 1, change the paragraph to read as follows:

63. Allowable Upper Bound: the highest value a system health performance counter can reach without violating the policy that monitors that counter. See "Upper Bound Tolerance". Glossary Item #59.

At page 20, line 5, change the paragraph to read as follows:

64. Allowable Lower Bound: the lowest value a system health performance counter can reach without violating the policy that monitors that counter. See "Lower Bound Tolerance". Glossary Item #58.

At page 20, line 9, change the paragraph to read as follows:

65. Saved Policy: a set of predefined XML statements describing a single health monitoring policy included with the HealthMonitor service. See "Unisys-Provided Policy File". Glossary Item #60.

At page 22, line 14, change the paragraph to read as follows:

80. APP Manager Product: see "NetIQ AppManager", Glossary Item # 30.

At page 22, line 29, change the paragraph to read as follows:

85. Global Health Events: see "Global HealthEvents", Glossary Item # 47.

At page 23, line 1, change the paragraph to read as follows:

86. Data points: see Glossary Item #67, "Data Points", and Glossary Item #42, "Current Data Point".

At page 23, line 11, change the paragraph to read as follows:

89. TCL: see "Trend Counter Limit (TCL)", Glossary Item #66.

At page 23, line 12, change the paragraph to read as follows:

90. Saved Policy: same as Glossary Item #65.

At page 24, line 6, change the paragraph to read as follows:

The exemplary Windows .NET service portion of the solution involved herein does the following: (Item "d" below is the focus of the present disclosure)

At page 25, line 20, change the paragraph to read as follows:

The final piece of the solution involves two new knowledge scripts (712) (Fig. 8) that run in the Server Director (an application in the Unisys Server 702) (Fig. 8). The scripts provide notification of health events as they are detected in an environment and a form that is already familiar to existing Server Sentinel customers. These scripts retrieve the server (non-predictive) alerts and predictive alerts from the event collections maintained by the HealthEvents object and flash the associated icon in the Server Director tree view to direct the end user to the site of the problem. (Predictive alerts will flash both the icon for the affected system component and the "Predictive Alerts" icon; server alerts will flash on the icon of the affected component; see Fig. 8.) Server Director also provides a rich set of additional corrective and notification actions that can be associated with either or both scripts, as the user desires.

At page 27, line 21, change the paragraph to read as follows:

When a complete sample set is collected, the algorithm calculates the average percent difference between each data point. The result is taken and multiplied by the size of the sample set to calculate the total percentage difference across the set. Because individual data values may be erratic, the total result for the set is compared against a lower and upper bound percent tolerance value. The algorithm also includes a flag that controls whether the code is looking for positive (increasing over time) trends or negative (decreasing over time) trends. For each sample set, the algorithm has three possible outcomes. The following scenario groups describes those outcomes when searching for an upward trend:

At page 29, line 23, change the paragraph to read as follows:

FIG. 8 illustrates a generalized block diagram in which the method of the present invention could be used. A Microsoft Windows .NET operating system exists (700), which communicates data to and from a Unisys server 702. A series of processes are included in module 720, which include a HealthEvents.dll file (706), which communicates with a data store 708. The data store 708 contains the PredictiveEvents and HealthEvents collections 710. The data in these collections is accessed through the HealthEvents.dll (706) hosted by the Microsoft Windows .NET operating system 700. This series of process and data ~~(720)~~ in module 720 receive their input from the HealthMonitor service 704. A user client application or script 712, also hosted by the Microsoft Windows .NET operating system 700, maintains communication with the set of processes in module 720 as well.

At page 35, line 4, change the paragraph to read as follows:

If A*N >U is true (Yes to inquiry D1), another inquiry is made at step D6 to check if C >U. C is the “common sense” value of glossary item #52. If C >U is false (No to inquiry D6), the process continues at Fig. 3 via bubble D7. If C >U is true (Yes to inquiry D6), the size of increase DeltaI (A*N first value in sample set) is calculated at step D8. Next, a process is executed to save DeltaI (Block D9), the “trend” counter is incremented (Block D10), and a process to zero the “not a trend” counter is initiated at step D11. An inquiry is then made at step D12 to check if the “trend” counter is greater than TCL. If the “trend” counter is not greater than TCL (No to inquiry D12), the process continues at Fig. 2 via bubble D5. If the “trend” counter is greater than TCL (Yes to inquiry D12), the trend is reported (Block D13), and all the counters are reset (Block D14), and the process continues at step D5 to go to Fig. 2.